PRESENTING SYSTEMS CONCEPTS IN PHYSIOLOGY AND PHARMACOLOGY WITH SIMULATION APPLETS IN JAVA

J.Mailen Kootsey, Grant McAuley, Hua Liu
Department of Physiology and Pharmacology,
Loma Linda University, Loma Linda, CA, USA

Abstract- **Java simulation applets solving equations for system models have been constructed for teaching system behavior in Pharmacology and Physiology. The applets are intended to be included in Web pages with text and other illustrations for use in the classroom or in self-study lessons. Students can experiment with the system by changing selected model parameters with sliders, immediately observing the resulting changes in system behavior. Several applet presentations are constructed for a model, each presentation designed for a different learning objective by displaying a subset of output variables and making a subset of parameters available for adjustment. Presentations may include control buttons, a graph for output display, sliders to adjust parameters, a legend table comparing parameter settings for multiple experiments, and an animation of the model linked to the calculations. The applet design is highly modular to facilitate replication with different models. Two architectures were tested and compared: 1) a single applet containing all the functions listed above embedded as a single unit in the Web page and 2) a cluster of individual applets of different functional types (control buttons, graph, sliders, etc.) distributed over the Web page and linked by a control object having no visible interface.**
***Keywords* - Computer simulation, mathematical model, teaching, Web pages, Java, applet, computer program, compartmental model**

I. INTRODUCTION

Faculty teaching complex biomedical systems such as compartmental models and cell regulation networks have traditionally used diagrams, equations, and graphs to illustrate system structure and behavior under different conditions. While the graphics communicate more than bare equations, the student must still synthesize a mental picture of dynamic behavior from static figures.

Computer simulation based on a mathematical system model can provide a direct experience with system dynamics and changing behavior under different external conditions[1]-[4]. Writing the code for such a simulation program is not a trivial undertaking, even for simple system models. In addition to the system equations, the code must include numerical solution for the equations, control of the simulation, provision for display of output (graph, animation, etc.), changing of parameter values, and comparison of output under different conditions.

Simulation software packages such as SCoP (Simulation Resources, Inc.), SAAM II (SAAM Institute, Inc.), and Stella (High Performance Systems, Inc.) are designed to simplify the process of building a simulation program[5][6]. However, these packages are designed to produce flexible programs to meet a wide variety of needs so access is provided to many services and settings. As a result, student using simulation programs built by these packages for learning exercises need training in the programs in order to be able to use them effectively. A teaching simulation should be clear enough to use without prior training and the options for change should be limited to focus the student's attention on a specific learning objective. Also, there is no clear way to combine simulations from these general packages with explanatory text and other illustrations or to deliver the simulations over the Web.

Java applets appear to be an excellent technology for building teaching simulations because applets can readily be combined with text, pictures, animations, and video clips for delivery over the Web. The applets can be viewed in standard browsers on most computer platforms. Improvements in Java technology have brought the computation speed up to be comparable to what can be achieved with FORTRAN or C/C++ compilers. Programming still remains a challenge, though.

Our goal in this project is to design a Java applet architecture for teaching purposes, meeting the following objectives:

1) The simulation should be self-contained for insertion into standard HTML Web pages and viewing on multiple platforms.
2) In addition to the model equations and numerical solution methods, each simulation should include computation controls, visual devices for changing parameter values, graphic output devices such as graphs and animations, and provision for the student to compare the outputs under different sets of conditions.
3) The number of controls available in any simulation should be limited to focus the student's attention on a single learning objective.
4) The architecture should make it easy to construct multiple simulations for a single system model, each emphasizing different learning objectives.
4) The software architecture should also make it easy to replace the system equations and numerical solver algorithm so that simulations can be built for a variety of systems.

II. METHODS

Applets in this project were implemented and tested using the Java 2 language with its Swing framework (Sun Microsystems) and the JBuilder 4 interactive development environment (Borland). We used the JClass JChart library (Sitraka) to build the output display graph. All applets were tested in both Netscape (version 4.75) and Internet Explorer

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25 Oct 2001	N/A	-

Title and Subtitle Presenting Systems Concepts in Physiology and Pharmacology With Simulation Applets in JAVA	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Department of Physiology and Pharmacology Loma Linda University Loma Linda, CA	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes Papers from 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 4	

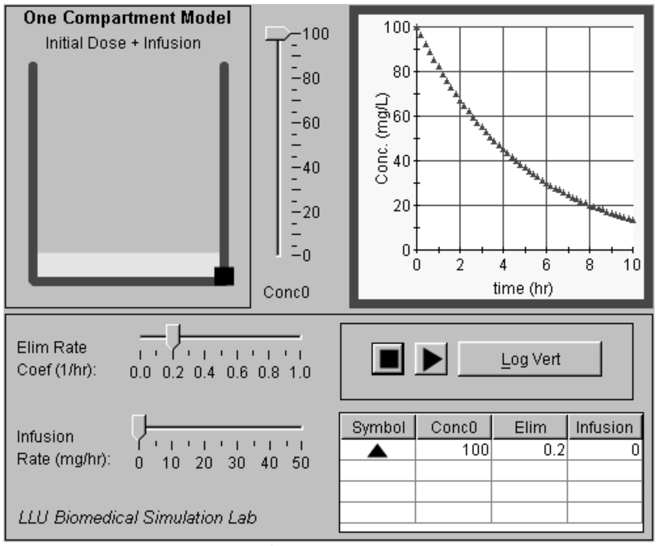

Figure 1. Teaching simulation for a one-compartment model, constructed as a single applet.

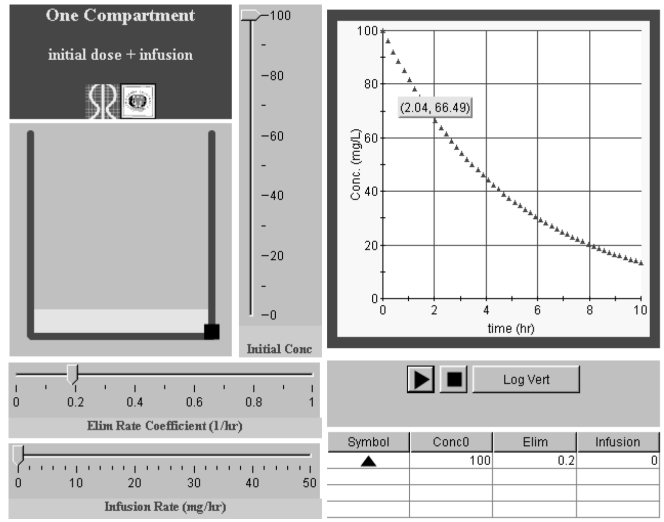

Figure 2. Teaching simulation for a one-compartment model, constructed as an applet cluster.

(version 5.5) browsers using the plug-in available from Sun Microsystems. Web pages were designed with either Dreamweaver (Macromedia) or Golive (Adobe) HTML editors.

III. Results

A. User Interface Features

Figs. 1 and 2 show simulations for a one-compartment model. In Fig. 1, the architecture is a single applet, while in Fig. 2, the simulation is a cluster of communicating applets. The same functionality is available in both forms.

To control the simulations, we adopted the symbols used by CD players to label the control buttons. The Run button is marked with a solid black triangle, pointing right (cf. “Play”). The Reset button is marked with a solid black square (cf. “Rewind”). The model calculations are automatically reset after each calculation run, so the function of the Reset button is to clear the graph and the graph legend table.

Each simulation has up to three sliders for adjusting parameters. Each slider is given a range appropriate to the parameter so the student cannot get erroneous output by putting in impossible parameter values, such as a negative rate coefficients for a reaction. The full mathematical model for any system includes a description of inputs to the system (stimuli), so the parameters available for adjustment include input controls as well as coefficients in the model equations.

The graph module can display up to four output variables with one or two vertical axis scales. Colored symbols are used to distinguish the variables in the graph, with the colors keyed to shapes in the animation. For example, in a two-compartment model (not shown), the two containers are blue and green in the animation, with the same colors used to plot drug concentrations from the two compartments. When multiple output variables are displayed, additional toggle buttons of matching color are added to the control panel to turn individual curves on and off in the graph.

We chose to plot the output for each variable as individual, unconnected points, emphasizing the discrete nature of both the simulation and of recorded data – even though the model variables are continuous. In a series of simulations, different symbols are used to distinguish the curves for each calculation run. A graph legend table associates the symbol shape with the parameter settings used to generate that output. A maximum of four simulations can be run before the student must click on the Reset button to clear the graph. This system of symbol shapes and legend table helps the student to study the change in model behavior resulting from parameter changes.

Since many biomedical models have distinctive output curve shapes in semi-logarithmic plots, we made provision for the student to switch back and forth between linear and semi-logarithmic modes. A toggle button in the control panel labeled “Log Vert” switches the vertical axis to a logarithmic scale. A second click on this button returns the scale to linear mode.

B. Animations

The animated graphic of the system being simulated adds significantly to the clarity of the simulation. For the one- and two-compartmental models, we drew a cross section of an open-topped container (beaker?) and animated the level of “fluid” in the container to follow the drug concentration as it changes in time. The elimination path is shown as a hole at the bottom of the container wall. The size of this hole varies as the student changes the elimination rate coefficient. In the two-compartment model, the “pipe” linking the two compartments also changes in diameter as the student changes the coupling rate coefficient.

The animations for the compartmental models were programmed directly in Java using the Java2D library. Because of the simple design possible for the compartment graphic, the programming effort required to produce the animation was not overwhelming. For more complex models such as the Hodgkin-Huxley nerve model, however, building a controllable animation is a formidable task. Several commercial programs are available for building animations and we are hoping to locate one that can build animations subject to control by external variables. Most animation programs produce a fixed motion sequence with the only possible variation being the playback speed.

C. Code Architecture

We tested two different architectures for the simulations: a single applet and a cluster of coupled applets. In the first architecture, all functions – computation, control, parameter variation, output display – are contained in a single applet with a fixed user interface. In the second architecture, each applet contains only one functional device such as a parameter slider, group of control buttons, graph, or animation. All the applets in the cluster interact with an "invisible" context manager object having no user interface. This context manager is a singleton object, meaning that only one can exist at any time.

The advantage of the single applet architecture is simplicity of insertion into a Web page. A single applet call brings in the entire simulation with control and output features. The price paid for this simplicity is inflexible visual design. Changes in layout or in the visual devices used call for a substantial reprogramming effort.

The cluster architecture has the opposite characteristics. The layout is done in the HTML page, not in the applet, so the layout can be changed in an HTML editor without reprogramming the Java code. Also, simulation controls and output displays can be mixed in with tutorial text and graphics, whereas with the single applet design, the text has to be kept separate from the simulation. The disadvantage of the cluster architecture is the increased work required to add the simulation to an HTML page. Multiple applets have to be specified with their parameters and the layout has to be designed in HTML.

In both architectures, we utilized object-oriented design for the user interface and the calculations. All control and output devices are described in classes so it is easy to substitute one visual device with another of equal functionality. For example, a numerical input or knob can easily replace a slider for parameter control. The model equations (sets of ordinary differential equations in the examples built so far) are contained in a single class and the numerical solver algorithm is also contained in one class. After constructing a series of simulations for one- and two-compartment models, we built a series of five simulation views based on the Hodgkin-Huxley nerve model. It took one programmer less than a day to replace the model equations and set up the graph and parameter sliders for the new model. We hope to reduce this reprogramming time further in the future with refinement of the Java classes

D. Applications

The simulations of either architecture can be placed in plain HTML pages and used as demonstrations in the classroom. We built a series of simulations of one- and two-compartment models and have used them in teaching pharmacokinetics to second-year medical students. A similar set of simulations of the Hodgkin-Huxley nerve model has been used in a neuroscience class for graduate students. In both cases, student reaction was positive and they requested a version they could download and experiment with themselves.

We have also written a tutorial on compartmental models in pharmacokinetics using the simulation applets along with text and static illustrations. Fig. 3 shows one screen from this tutorial. The screen area was divided into four panels: a narrow upper panel for titles, a large central panel, a vertical side panel, and a small panel for navigation buttons. In the tutorial, the central panel sometimes contains the simulation and at other times it is used for text and static graphics. The side panel contains a table of contents at first and later displays instructional text when the central panel contains a simulation.

The combination of HTML pages with embedded Java applets permits easy distribution from a Web server or on a CDROM. The archive (jar) file containing the Java classes for a simulation must be downloaded at the start of a lesson. The size of this file for the compartmental model tutorial is a little over a half megabyte, resulting in a download time of several minutes for a typical dial-up modem connection.

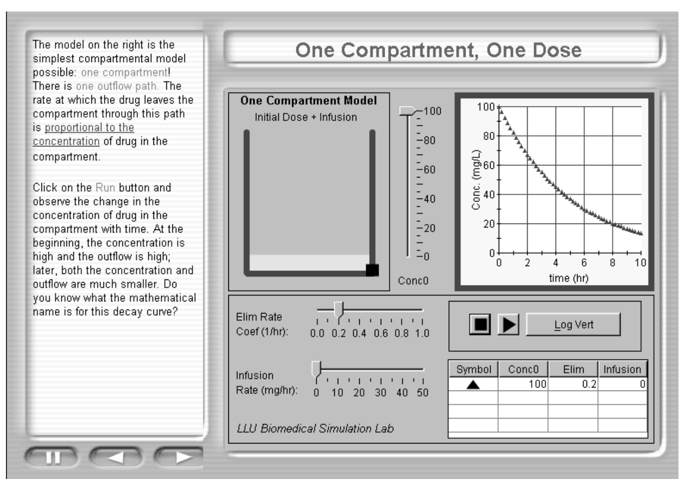

Figure 3. One page of a tutorial on compartmental models, utilizing the simulation constructed as a single applet.

IV. Conclusions

The goals for this project were met by both software architectures – single applet and applet cluster. We will continue to test both forms, but we expect the versatility of the applet cluster architecture to win out over the simplicity of the single applet in the long run. With additional refinement of the Java class structure, we can probably simplify still further the tasks of inserting a new model and creating different views of a model (plotting different output variables and exposing different parameter subsets for variation). At the present time, the most difficult step in building one of these interactive teaching simulations is creating the animation to be driven by the output variables. Hopefully, a way can be found in the future to simplify and speed this step as the animation can make a major contribution to the understanding available through the simulation.

References

[1] T.M. Dwyer, J. Fleming, J.E. Randall, and T.G. Coleman, "Teaching physiology and the World Wide Web: Electrochemistry and electrophysiology on the Internet," *Adv. Physiol. Educ.*, vol. 18, pp. S2-S13, 1997.

[2] E. Nikkelen, W.L. van Meurs, and M.A.K. Ohrn, "Hydraulic analog for simultaneous representation of pharmacokinetics and pharmacodynamics: Application to vecuronium," *J. Clin. Mon. Comput.*, vol. 14, pp. 329-337, 1998.

[3] M. Haag, L. Maylein, F.J. Leven, B. Tonshoff, and R. Haux, "Web-based training: a new paradigm in computer-assisted instruction in medicine," *Int. J. Med. Inform.*, vol. 53, pp. 79-90, 1999.

[4] R. Mehvar, "Development and application of an on-line modele for teaching Bayesian forecasting principles in a clinical pharmacokinetics course," *Am. J. Pharm. Educ.*, vol. 64, pp. 121-125, 2000.

[5] M.C. Kohn, J.M. Kootsey, and M.D. Feezor, "A block organized model builder", *Mathl. Comput. Modeling*, vol. 19, pp. 75-97, 1994.

[6] A. Vawer and J. Rashbass, "The biological toolbox: A computer program for simulating basic biological and pathological processes," *Comp. Meth. Prog. Biomed.*, vol. 52, pp. 203-211, 1997.